Cutting Through Regular Post Embedding Problems*

P. Karandikar[1] and Ph. Schnoebelen[2]

[1] Chennai Mathematical Institute
[2] LSV, ENS Cachan, CNRS

Abstract. The Regular Post Embedding Problem extended with partial (co)directness is shown decidable. This extends to universal and/or counting versions. It is also shown that combining directness and codirectness in Post Embedding problems leads to undecidability.

1 Introduction

The Regular Post Embedding Problem (PEP for short, named by analogy with Post's Correspondence Problem) is the problem of deciding, given two morphisms on words $u, v : \Sigma^* \to \Gamma^*$ and a regular language $R \in \mathsf{Reg}(\Sigma)$, whether there is $\sigma \in R$ such that $u(\sigma)$ is a (scattered) subword of $v(\sigma)$. The *subword* ordering, also called *embedding*, is denoted "$\sqsubseteq$": $u(\sigma) \sqsubseteq v(\sigma) \overset{\text{def}}{\Leftrightarrow} u(\sigma)$ can be obtained by erasing some letters from $v(\sigma)$, possibly all of them, possibly none. Equivalently, PEP is the question whether a rational relation, or a transduction, $T \subseteq \Gamma^* \times \Gamma^*$ intersects non-vacuously the subword relation, hence is a special case of the intersection problem for two rational relations.

This problem is new and quite remarkable: it is decidable [2] but surprisingly hard since it is not primitive-recursive and not even multiply-recursive. In fact, it is at level $\mathcal{F}_{\omega^\omega}$ (and not below) in the Fast-Growing Hierarchy [8, 12].

A variant problem, $\mathsf{PEP}_{\text{dir}}$, asks for the existence of *direct* solutions, i.e., solutions $\sigma \in R$ such that $u(\tau) \sqsubseteq v(\tau)$ for every prefix τ of σ. The two problems are inter-reducible [4], hence have the same complexity: decidability of PEP entails decidability of $\mathsf{PEP}_{\text{dir}}$, while hardness of $\mathsf{PEP}_{\text{dir}}$ entails hardness for PEP.

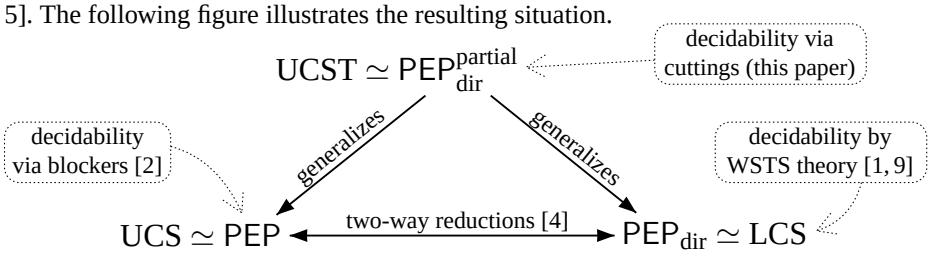
Our contribution. We introduce $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$, or "PEP with *partial* directness", a new problem that generalizes both PEP and $\mathsf{PEP}_{\text{dir}}$, and prove its decidability. The proof combines two ideas. Firstly, by Higman's Lemma, a long solution must eventually contain "*comparable*" so-called cutting points, from which one deduces that the solution is not minimal (or unique, or ...). Secondly, the above notion of "*eventually*", that comes from Higman's Lemma, can be turned into an effective upper bound thanks to a Length Function Theorem. The cutting technique described above was first used in [7] for reducing $\exists^\infty$PEP to PEP. In this paper we use it to obtain a decidability proof for $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ that is not only more general but also more direct than the earlier proofs for PEP or $\mathsf{PEP}_{\text{dir}}$. It also immediately provides an $\mathcal{F}_{\omega^\omega}$ complexity upper bound. We also

* Supported by ARCUS 2008–11 Île de France-Inde and Grant ANR-11-BS02-001. The first author was partially funded by Tata Consultancy Services.

show the decidability of universal and/or counting versions of the extended $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ problem, and explain how our attempts at further generalisation, most notably by considering the combination of directness and codirectness in a same instance, lead to undecidability.

Applications to channel machines. Beyond the tantalizing decidability questions, our interest in PEP and its variants comes from their close connection with fifo channel machines [11], a family of computational models that are a central tool in several areas of program and system verification (see [5] and the references therein). Here, PEP and its variants provide abstract versions of verification problems for channel machines [4], bringing greater clarity and versatility in both decidability and undecidability (more generally, hardness) proofs.

Beyond providing a uniform and simpler proof for the decidability of PEP and $\mathsf{PEP}_{\text{dir}}$, our motivation for considering $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is that it allows solving the decidability of UCST, i.e., unidirectional channel systems (with one reliable and one lossy channel) *extended with the possibility of testing the contents of channels* [10]. We recall that PEP was introduced for UCS, unidirectional channel systems where tests on channels are not supported [4, 3], and that $\mathsf{PEP}_{\text{dir}}$ corresponds to LCS, i.e., lossy channel systems, for which verification is decidable using techniques from WSTS theory [1, 9, 5]. The following figure illustrates the resulting situation.



Outline of the paper. Section 2 recalls basic notations and definitions. In particular, it explains the Length Function Theorem for Higman's Lemma, and lists basic results where the subword relation interacts with concatenations and factorization. Section 3 contains our main result, a direct decidability proof for $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$, a problem subsuming both PEP and $\mathsf{PEP}_{\text{dir}}$. Section 4 builds on this result and shows the decidability of counting problems on $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$. Section 5 further shows the decidability of universal variants of these questions. Section 6 contains our undecidability results for extensions of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$. Proofs omitted for lack of space can be found in the full version of this paper, available at arxiv.org/abs/1109.1691.

2 Basic notation and definitions

Words. Concatenation of words is denoted multiplicatively, with ε denoting the empty word. If s is a prefix of a word t, $s^{-1}t$ denotes the unique word s' such that $t = ss'$, and $s^{-1}t$ is not defined if s is not a prefix of t. Similarly, when s is a suffix of t, ts^{-1} is t with the s suffix removed. For a word $x = \mathtt{a}_0 \dots \mathtt{a}_{n-1}$, $\tilde{x} \stackrel{\text{def}}{=} \mathtt{a}_{n-1} \dots \mathtt{a}_0$ is the mirrored word.

The mirror of a language R is $\tilde{R} \stackrel{\text{def}}{=} \{\tilde{x} \mid x \in R\}$. We write $s \sqsubseteq t$ when s is a subword (subsequence) of t.

Lemma 2.1 (Subwords and concatenation). *For all words y,z,s,t:*
1. If $yz \sqsubseteq st$ then $y \sqsubseteq s$ or $z \sqsubseteq t$.
2. If $yz \sqsubseteq st$ and $z \sqsubseteq t$ and x is the longest suffix of y such that $xz \sqsubseteq t$, then $yx^{-1} \sqsubseteq s$.
3. If $yz \sqsubseteq st$ and $z \not\sqsubseteq t$ and x is the shortest prefix of z such that $x^{-1}z \sqsubseteq t$, then $yx \sqsubseteq s$.
4. If $yz \sqsubseteq st$ and $z \sqsubseteq t$ and x is the longest prefix of t such that $z \sqsubseteq x^{-1}t$, then $y \sqsubseteq sx$.
5. If $yz \sqsubseteq st$ and $z \not\sqsubseteq t$ and x is the shortest suffix of s such that $z \sqsubseteq xt$, then $y \sqsubseteq sx^{-1}$.
6. If $sx \sqsubseteq yt$ and $t \sqsubseteq s$, then $sx^k \sqsubseteq y^k t$ for all $k \geq 1$.
7. If $xs \sqsubseteq ty$ and $t \sqsubseteq s$, then $x^k s \sqsubseteq ty^k$ for all $k \geq 1$.

With a language R one associates a congruence (wrt concatenation) given by $s \sim_R t \stackrel{\text{def}}{\Leftrightarrow} \forall x,y(xsy \in R \Leftrightarrow xty \in R)$ and called the syntactic congruence (also, the syntactic monoid). This congruence has finite index if (and only if) R is regular. For regular R, let n_R denote this index: $n_R \leq m^m$ when R is recognized by a m-state complete DFA.

Higman's Lemma. It is well-known that for words over a finite alphabet, $\sqsubseteq$ is a well-quasi-ordering, that is, any infinite sequence of words $x_1, x_2, x_3, \ldots$ contains an infinite increasing subsequence $x_{i_1} \sqsubseteq x_{i_2} \sqsubseteq x_{i_3} \sqsubseteq \cdots$. This result is called Higman's Lemma.

For $n \in \mathbb{N}$, we say that a sequence (finite or infinite) of words is n-good if it has an increasing subsequence of length n. It is n-bad otherwise. Higman's Lemma tells us that every infinite sequence is n-good for every n. Hence every n-bad sequence is finite.

It is often said that Higman's Lemma is "non-effective" since it does not give any explicit information on the maximal length of bad sequences. Consequently, when one uses Higman's Lemma to prove that an algorithm terminates, no meaningful upper-bound on the algorithm's running time is derived from the proof. However, complexity upper-bound can be derived if the complexity of the sequences (or more precisely of the process that generates bad sequences) is taken into account. The interested reader can consult [12] for more details. Here we only need the simplest version of these results, i.e., the statement that the maximal length of bad sequences is computable.

A sequence of words $x_1, \ldots, x_l$ is *k-controlled* ($k \in \mathbb{N}$) if $|x_i| \leq ik$ for all $i = 1, \ldots, l$.

Length Function Theorem. *There exists a computable function $H : \mathbb{N}^3 \to \mathbb{N}$ such that any n-bad k-controlled sequences of words in Γ^* has length at most $H(n,k,|\Gamma|)$. Furthermore, H is monotonic in all three arguments.*

Thus, a sequence with more than $H(n,k,|\Gamma|)$ words is n-good or is not k-controlled. We refer to [12] for the complexity of H. Here it is enough to know that H is computable.

3 Deciding $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$, or PEP with partial directness

We introduce $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$, a problem generalizing both PEP and $\mathsf{PEP}_{\text{dir}}$, and show its decidability. This is proved by showing that if a $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ instance has a solution, then it has a solution whose length is bounded by a computable function of the input. This proof is simpler and more direct than the proof (for PEP only) based on blockers [2].

Definition 3.1. $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ *is the problem of deciding, given morphisms* $u,v:\Sigma^*\to\Gamma^*$ *and regular languages* $R,R'\in\mathsf{Reg}(\Sigma)$*, whether there is* $\sigma\in R$ *such that* $u(\sigma)\sqsubseteq v(\sigma)$ *and* $u(\tau)\sqsubseteq v(\tau)$ *for all prefixes* τ *of* σ *belonging to* R' *(in which case* σ *is called* a solution*).* $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ *is the variant problem of deciding whether there is* $\sigma\in R$ *such that* $u(\sigma)\sqsubseteq v(\sigma)$ *and* $u(\tau)\sqsubseteq v(\tau)$ *for all suffixes of* τ *of* σ *that belong to* R'*.*

Both PEP and $\mathsf{PEP}_{\text{dir}}$ are special cases of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$, obtained by taking $R'=\varnothing$ and $R'=\Sigma^*$ respectively. Obviously $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ and $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ are two equivalent presentations, modulo mirroring, of a same problem. Given a $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ (or $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$) instance, we let $K_u\stackrel{\text{def}}{=}\max_{a\in\Sigma}|u(a)|$ denote the *expansion factor* of u and say that $\sigma\in\Sigma^*$ is *long* if $|\sigma|>2H(n_Rn_{R'}+1,K_u,|\Gamma|)$, otherwise it is *short* (recall that $H(n,k,|\Gamma|)$ was defined with the Length Function Theorem). In this section we prove:

Theorem 3.2. *A* $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ *or* $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ *instance has a solution if, and only if, it has a short solution. This entails that* $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ *and* $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ *are decidable.*

Decidability is an obvious consequence since the maximal length for short solutions is computable, and since it is easy to check whether a candidate σ is a solution. Furthermore, one derives an upper bound on the complexity of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ since the Length Function H is bounded in $\mathcal{F}_{\omega^\omega}$ [12].

For the proof of Theorem 3.2, we find it easier to reason on the codirect version. Pick an arbitrary $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance (Σ,Γ,u,v,R,R') and a solution σ. Write $N=|\sigma|$ for its length, $\sigma[0,i)$ and $\sigma[i,N)$ for, respectively, its prefix of length i and its suffix of length $N-i$. Two indices $i,j\in[0,N]$ are *congruent* if $\sigma[i,N)\sim_R\sigma[j,N)$ and $\sigma[i,N)\sim_{R'}\sigma[j,N)$. When σ is fixed, as in the rest of this section, we use shorthand notations like $u_{0,i}$ and $v_{i,j}$ to denote the images, here $u(\sigma[0,i))$ and $v(\sigma[i,j))$, of factors of σ.

We prove two "cutting lemmas" giving sufficient conditions for "cutting" a solution $\sigma=\sigma[0,N)$ along certain indices $a<b$, yielding a shorter solution $\sigma'=\sigma[0,a)\sigma[b,N)$. Here the following notation is useful. We associate, with every suffix τ of σ', a corresponding suffix, denoted $S(\tau)$, of σ: if τ is a suffix of $\sigma[b,N)$, then $S(\tau)\stackrel{\text{def}}{=}\tau$, otherwise, $\tau=\sigma[i,a)\sigma[b,N)$ for some $i<a$ and we let $S(\tau)\stackrel{\text{def}}{=}\sigma[i,N)$. In particular $S(\sigma')=\sigma$.

An index $i\in[0,N]$ is said to be *blue* if $u_{i,N}\sqsubseteq v_{i,N}$, it is *red* otherwise. In particular, N is blue trivially, 0 is blue since σ is a solution, and i is blue whenever $\sigma[i,N)\in R'$. If i is a blue index, let $l_i\in\Gamma^*$ be the longest suffix of $u_{0,i}$ such that $l_iu_{i,N}\sqsubseteq v_{i,N}$ and call it the *left margin* at i.

Lemma 3.3 (Cutting lemma for blue indices). *Let* $a<b$ *be two congruent and blue indices. If* $l_a\sqsubseteq l_b$*, then* $\sigma'=\sigma[0,a)\sigma[b,N)$ *is a solution (shorter than* σ*).*

Proof. Clearly $\sigma'\in R$ since $\sigma\in R$ and a and b are congruent. Also, for all suffixes τ of σ', $S(\tau)\in R'$ iff $\tau\in R'$.

We claim that, for any suffix τ of σ', if $u(S(\tau))\sqsubseteq v(S(\tau))$ then $u(\tau)\sqsubseteq v(\tau)$. This is obvious when $\tau=S(\tau)$, so we assume $\tau\neq S(\tau)$, i.e., $\tau=\sigma[i,a)\sigma[b,N)$ and $S(\tau)=\sigma[i,N)$ for some $i<a$. Assume $u(S(\tau))\sqsubseteq v(S(\tau))$, i.e., $u_{i,N}\sqsubseteq v_{i,N}$. Now at least one of $u_{i,a}$ and l_a is a suffix of the other, which gives two cases. If $u_{i,a}$ is a suffix of l_a, then

$$u(\tau)=u_{i,a}u_{b,N}\sqsubseteq l_au_{b,N}\sqsubseteq l_bu_{b,N}\sqsubseteq v_{b,N}\sqsubseteq v(\tau)\,.$$

Otherwise, $u_{i,a} = xl_a$ for some x (see Fig. 1). Then $u_{i,N} \sqsubseteq v_{i,N}$ rewrites as $u_{i,a}u_{a,N} =$

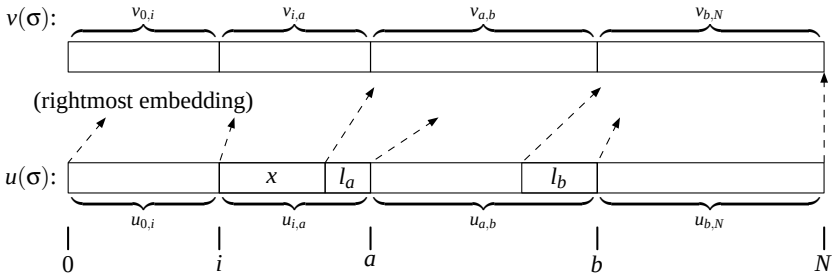


Fig. 1. Schematics for Lemma 3.3, with $l_a \sqsubseteq l_b$

$xl_au_{a,N} \sqsubseteq v_{i,a}v_{a,N}$. Now, and since l_a is the longest suffix for which $l_au_{a,N} \sqsubseteq v_{a,N}$, Lemma 2.1.2 entails $x \sqsubseteq v_{i,a}$. Combining with $l_a \sqsubseteq l_b$ (assumption of the Lemma) gives:

$$u(\tau) = u_{i,a}u_{b,N} = xl_au_{b,N} \sqsubseteq v_{i,a}l_bu_{b,N} \sqsubseteq v_{i,a}v_{b,N} = v(\tau)\,.$$

This shows that σ' is a solution (which completes the proof) since we can infer $u(\tau) \sqsubseteq v(\tau)$ for any suffix $\tau \in R'$ (or for $\tau = \sigma'$) from the corresponding $u(S(\tau)) \sqsubseteq v(S(\tau))$. □

If i is a red index, let $r_i \in \Gamma^*$ be the shortest prefix of $u_{i,N}$ such that $r_i^{-1}u_{i,N} \sqsubseteq v_{i,N}$ (equivalently $u_{i,N} \sqsubseteq r_iv_{i,N}$) and call it the *right margin* at i.

Lemma 3.4 (Cutting lemma for red indices). *Let $a < b$ be two congruent and red indices. If $r_b \sqsubseteq r_a$, then $\sigma' = \sigma[0,a)\sigma[b,N)$ is a solution (shorter than σ).*

Proof. Write $u_{b,N}$ under the form r_bx so that $x \sqsubseteq v_{b,N}$. We proceed as for Lemma 3.3 and show that $u(S(\tau)) \sqsubseteq v(S(\tau))$ implies $u(\tau) \sqsubseteq v(\tau)$ for all suffixes τ of σ'. Assume $u(S(\tau)) \sqsubseteq v(S(\tau))$ for some τ. The only interesting case is when $\tau \neq S(\tau)$ and $\tau = \sigma[i,a)\sigma[b,N)$ for some $i < a$ (see Fig. 2).

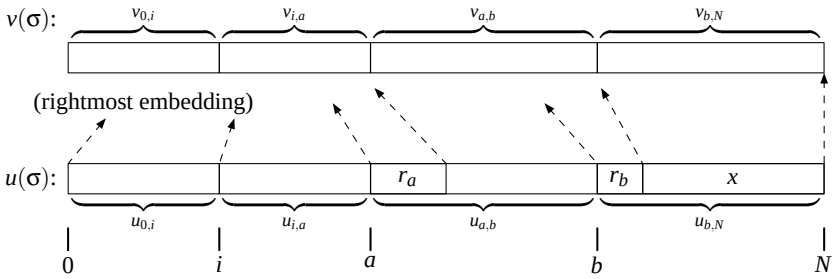


Fig. 2. Schematics for Lemma 3.4, with $r_b \sqsubseteq r_a$

From $u_{i,N} = u_{i,a}u_{a,N} \sqsubseteq v_{i,a}v_{a,N} = v_{i,N}$, i.e., $u(S(\tau)) \sqsubseteq v(S(\tau))$, and $u_{a,N} \not\sqsubseteq v_{a,N}$ (since a is a red index), the definition of r_a entails $u_{i,a}r_a \sqsubseteq v_{i,a}$ (Lemma 2.1.3). Then

$$u(\tau) = u_{i,a}u_{b,N} = u_{i,a}r_bx \sqsubseteq u_{i,a}r_av_{b,N} \sqsubseteq v_{i,a}v_{b,N} = v(\tau)\,. \qquad \square$$

We now conclude the proof of Theorem 3.2. Let $g_1 < g_2 < \cdots < g_{N_1}$ be the blue indices in σ, let $b_1 < b_2 < \cdots < b_{N_2}$ be the red indices, and look at the corresponding sequences $(l_{g_i})_{i=1,\ldots,N_1}$ of left margins and $(r_{b_i})_{i=1,\ldots,N_2}$ of right margins.

Lemma 3.5. *$|l_{g_i}| \leq (i-1) \times K_u$ for all $i = 1,\ldots,N_1$, and $|r_{b_i}| \leq (N_2 - i + 1) \times K_u$ for all $i = 1,\ldots,N_2$. In other words, the sequence on left margins and the* reversed *sequence of right margins are K_u-controlled.*

Now, let $N_c \stackrel{\text{def}}{=} n_R n_{R'} + 1$ and $L \stackrel{\text{def}}{=} H(N_c, K_u, |\Gamma|)$ and assume $N > 2L$. Since $N_1 + N_2 = N + 1$, either σ has at least $L+1$ blue indices and, by definition of L and H, there exist N_c blue indices $a_1 < a_2 < \cdots < a_{N_c}$ with $l_{a_1} \sqsubseteq l_{a_2} \sqsubseteq \cdots \sqsubseteq l_{a_{N_c}}$, or σ has at least $L+1$ red indices and there exist N_c red indices $a'_1 < a'_2 < \cdots < a'_{N_c}$ with $r_{a'_{N_c}} \sqsubseteq \cdots \sqsubseteq r_{a'_2} \sqsubseteq r_{a'_1}$ (since it is the reversed sequence of right margins that is controlled). Out of $N_c = 1 + n_R n_{R'}$ indices, two must be congruent, fulfilling the assumptions of Lemma 3.3 or Lemma 3.4. Therefore σ is not the shortest solution, proving Theorem 3.2.

4 Counting the number of solutions

We consider two counting questions: $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is the question whether a $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ instance has infinitely many solutions (a decision problem), while $\#\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is the problem of computing the number of solutions of the instance (a number in $\mathbb{N} \cup \{\infty\}$). For technical convenience, we often deal with the (equivalent) codirected versions, $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\#\mathsf{PEP}^{\text{partial}}_{\text{codir}}$.

For an instance $(\Sigma, \Gamma, u, v, R, R')$, we let $K_v \stackrel{\text{def}}{=} \max_{a \in \Sigma} |v(a)|$ and define

$$L \stackrel{\text{def}}{=} H(n_R n_{R'} + 1, K_v, |\Gamma|)\,, \qquad L' \stackrel{\text{def}}{=} H\Big(\big[n_R(2L+1)\big]^{n_R(2L+1)} n_{R'} + 1, K_u, |\Gamma|\Big)\,.$$

We say that a solution $\sigma \in \Sigma^*$ is *long* if $|\sigma| > 2L$ and *very long* if $|\sigma| > 2L'$ (note that "long" is slightly different from "not short" from Section 3). In this section we prove:

Theorem 4.1. *For a $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ or $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance, the following are equivalent:*
(a). It has infinitely many solutions.
(b). It has a long solution.
(c). It has a solution that is long but not very long.

From this, it will be easy to count the number of solutions:

Corollary 4.2. *$\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ and $\exists^{\infty}\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ are decidable, $\#\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ and $\#\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ are computable.*

Proof. Decidability for the decision problems is clear since L and L' are computable.

For actually counting the solutions, we check whether the number of solutions is finite or not using the decision problems. If infinite, we are done. If finite, we first compute an upper bound on the length of the longest solution. For this we build $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ (resp. $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$) instances where R is replaced by $R \smallsetminus \Sigma^{\leq M}$ (which is regular when R is) for increasing values of $M \in \mathbb{N}$. When eventually M is large enough, the instance is negative and this can be detected (by Theorem 3.2). Once we know that there are no solutions longer than M, counting solutions is done by finite enumeration. □

We now prove Theorem 4.1. First observe that if the instance has a long solution, it has a solution with R replaced by $R \cap \Sigma^{>2L}$. This language has a DFA with $n_R(2L+1)$ states, thus the associated congruence has index at most $(n_R(2L+1))^{n_R(2L+1)}$. From Theorem 3.2, the instance has a solution which is long but not very long. Hence (b) and (c) are equivalent.

It remains to show (b) implies (a) since obviously (a) implies (b). For this we fix an arbitrary $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance $(\Sigma, \Gamma, u, v, R, R')$ and consider a solution σ, of length N. We develop two so-called "iteration lemmas" that are similar to the cutting lemmas from Section 3, with the difference that they expand σ instead of reducing it.

As before, an index $i \in [0,N]$ is said to be *blue* if $u_{i,N} \sqsubseteq v_{i,N}$, and *red* otherwise. With blue and red indices we associate words analogous to the l_i's and r_i's from Section 3, however now they are factors of $v(\sigma)$, not $u(\sigma)$ (hence the different definition for L). The terms "left margin" and "right margin" will be reused here for these factors.

We start with blue indices. For a blue index $i \in [0,N]$, let s_i be the longest prefix of $v_{i,N}$ such that $u_{i,N} \sqsubseteq s_i^{-1} v_{i,N}$ (equivalently $s_i u_{i,N} \sqsubseteq v_{i,N}$) and call it the *right margin* at i.

Lemma 4.3. *Suppose $a < b$ are two blue indices with $s_b \sqsubseteq s_a$. Then for all $k \geq 1$, $s_a (u_{a,b})^k \sqsubseteq (v_{a,b})^k s_b$.*

Proof. $s_a u_{a,N} \sqsubseteq v_{a,N}$ expands as $(s_a u_{a,b}) u_{b,N} \sqsubseteq v_{a,b} v_{b,N}$. Since $u_{b,N} \sqsubseteq v_{b,N}$, Lemma 2.1.4 yields $s_a u_{a,b} \sqsubseteq v_{a,b} s_b$. One concludes with Lemma 2.1.6, using $s_b \sqsubseteq s_a$. □

Lemma 4.4 (Iteration lemma for blue indices). *Let $a < b$ be two congruent and blue indices. If $s_b \sqsubseteq s_a$, then for every $k \geq 1$, $\sigma' = \sigma[0,a).\sigma[a,b)^k.\sigma[b,N)$ is a solution.*

Now to red indices. For a red index $i \in [0,N]$, let t_i be the shortest suffix of $v_{0,i}$ such that $u_{i,N} \sqsubseteq t_i v_{i,N}$. This is called the *left margin* at i. Thus, for a blue j such that $j < i$, $u_{j,N} \sqsubseteq v_{j,N}$ implies $u_{j,i} t_i \sqsubseteq v_{j,i}$ by Lemma 2.1.5.

Lemma 4.5 (Iteration lemma for red indices). *Let $a < b$ be two congruent and red indices. If $t_a \sqsubseteq t_b$, then for every $k \geq 1$, $\sigma' = \sigma[0,a).\sigma[a,b)^k.\sigma[b,N)$ is a solution.*

We now conclude the proof of Theorem 4.1. We first prove that the $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ instance has infinitely many solutions iff it has a long solution. Obviously, only the right-to-left implication has to be proven.

Suppose there are N_1 blue indices in σ, say $g_1 < g_2 < \cdots < g_{N_1}$; and N_2 red indices, say $b_1 < b_2 < \cdots < b_{N_2}$.

Lemma 4.6. *$|s_{g_i}| \leq (N_1 - i + 1) \times K_v$ for all $i = 1, \ldots, N_1$, and $|t_{b_i}| \leq (i-1) \times K_v$ for all $i = 1, \ldots, N_2$. That is, the* reversed *sequence of right margins and the sequence of left margins are K_v-controlled.*

Assume that σ is a long solution of length $N \geq 2L+1$. At least $L+1$ indices among $[0,N]$ are blue, or at least $L+1$ are red. We apply one of the two above claims, and from either $s_{g_{N_1}}, \ldots, s_{g_1}$ (if $N_1 \geq L+1$) or $t_{b_1}, \ldots, t_{b_{N_2}}$ (if $N_2 \geq L+1$) we get an increasing subsequence of length $n_R n_{R'} + 1$. Among these there must be two congruent indices. Then we get infinitely many solutions by Lemma 4.4 or Lemma 4.5.

5 Universal variants of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$

We consider universal variants of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ (or rather $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ for the sake of uniformity). Formally, given instances $(\Sigma, \Gamma, u, v, R, R')$ as usual, $\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is the question whether *every* $\sigma \in R$ is a solution, i.e., satisfies both $u(\sigma) \sqsubseteq v(\sigma)$ and $u(\tau) \sqsubseteq v(\tau)$ for all suffixes τ that belong to R'. Similarly, $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is the question whether "almost all", i.e., *all but finitely many*, σ in R are solutions, and $\#\neg\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is the associated counting problem that asks how many $\sigma \in R$ are not solutions.

The special cases $\forall\mathsf{PEP}$ and $\forall^\infty\mathsf{PEP}$ (where $R' = \varnothing$) have been shown decidable in [7] where it appears that, at least for Post Embedding, universal questions are simpler than existential ones. We now observe that $\forall\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ and $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ are easy to solve too: partial codirectness constraints can be eliminated since universal quantifications commute with conjunctions (and since the codirectness constraint is universal itself).

Lemma 5.1. *$\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ many-one reduce to $\forall^\infty\mathsf{PEP}$.*

Corollary 5.2. *$\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ and $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ are decidable, $\#\neg\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ is computable.*

We now prove Lemma 5.1. First, $\forall\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ easily reduces to $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$: add an extra letter $\mathtt{z}$ to Σ with $u(\mathtt{z}) = v(\mathtt{z}) = \varepsilon$ and replace R and R' with $R.\mathtt{z}^*$ and $R'.\mathtt{z}^*$. Hence the second half of the lemma entails its first half by transitivity of reductions.

For reducing $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$, it is easier to start with the negation of our question:

$$\exists^\infty\sigma \in R : \big(u(\sigma) \not\sqsubseteq v(\sigma) \text{ or } \sigma \text{ has a suffix } \tau \text{ in } R' \text{ with } u(\tau) \not\sqsubseteq v(\tau)\big)\,. \tag{$*$}$$

Call $\sigma \in R$ a *type 1 witness* if $u(\sigma) \not\sqsubseteq v(\sigma)$, and a *type 2 witness* if it has a suffix $\tau \in R'$ with $u(\tau) \not\sqsubseteq v(\tau)$. Statement $(*)$ holds if, and only if, there are infinitely many type 1 witnesses or infinitely many type 2 witnesses. The existence of infinitely many type 1 witnesses (call that "case 1") is the negation of a $\forall^\infty\mathsf{PEP}$ question. Now suppose that there are infinitely many type 2 witnesses, say $\sigma_1, \sigma_2, \ldots$ For each i, pick a suffix τ_i of σ_i such that $\tau_i \in R'$ and $u(\tau_i) \not\sqsubseteq v(\tau_i)$. The set $\{\tau_i \mid i = 1, 2, \ldots\}$ of these suffixes can be finite or infinite. If it is infinite ("case 2a"), then

$$u(\tau) \not\sqsubseteq v(\tau) \text{ for infinitely many } \tau \in (\overrightarrow{R} \cap R')\,, \tag{$**$}$$

where $\overrightarrow{R}$ is short for $\overrightarrow{^{\geq 0}R}$ and for $k \in \mathbb{N}$, $\overrightarrow{^{\geq k}R} \stackrel{\text{def}}{=} \{y \mid \exists x : (|x| \geq k \text{ and } xy \in R)\}$ is the set of the suffixes of words from R one obtains by removing at least k letters. Observe that, conversely, $(**)$ implies the existence of infinitely many type 2 witnesses (for a proof, pick $\tau_1 \in \overrightarrow{R} \cap R'$ satisfying the above, choose $\sigma_1 \in R$ of which τ_1 is a suffix. Then choose τ_2 such that $|\tau_2| > |\sigma_1|$, and proceed similarly).

On the other hand, if $\{\tau_i \mid i = 1, 2, \ldots\}$ is finite ("case 2b"), then there is a $\tau \in R'$ such that $u(\tau) \not\sqsubseteq v(\tau)$ and $\sigma'\tau \in R$ for infinitely many σ'. By a standard pumping argument, the second point is equivalent to the existence of some such σ' with also $|\sigma'| > k_R$, where k_R is the size of a NFA for R (taking $k_R = n_R$ also works). Write now $\hat{R}$ for $\overrightarrow{^{>k_R}R}$: if $\{\tau_i \mid i = 1, 2, \ldots\}$ is finite, then $u(\tau) \not\sqsubseteq v(\tau)$ for some τ in $(R' \cap \hat{R})$, and conversely this implies the existence of infinitely many type 2 witnesses.

To summarize, and since $\overrightarrow{R}$ and $\hat{R}$ are regular and effectively computable from R, we have just reduced $\forall^\infty\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ to the following conjunction

$$\underbrace{\forall^\infty\sigma\in R : u(\sigma)\sqsubseteq v(\sigma)}_{\text{not case 1}}\bigwedge\underbrace{\forall^\infty\tau\in(\overrightarrow{R}\cap R') : u(\tau)\sqsubseteq v(\tau)}_{\text{not case 2a}}\bigwedge\underbrace{\forall\tau\in(\hat{R}\cap R') : u(\tau)\sqsubseteq v(\tau)}_{\text{not case 2b}}.$$

This is now reduced to a single $\forall^\infty\mathsf{PEP}$ instance by rewriting the $\forall\mathsf{PEP}$ into a $\forall^\infty\mathsf{PEP}$ (as said in the beginning of this proof) and relying on distributivity:

$$\bigwedge_{i=1}^{n}\left[\forall^\infty x\in X_i : \dots \text{ some property } \dots\right] \equiv \forall^\infty x\in\bigcup_{i=1}^{n} X_i : \dots \text{ same } \dots .$$

6 Undecidability for $\mathsf{PEP}_{\text{co\&dir}}$ and other extensions

The decidability of $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is a non-trivial generalization of previous results for PEP. It is a natural question whether one can further generalize the idea of partial directness and maintain decidability. In this section we describe two attempts that lead to undecidability, even though they remain inside the regular PEP framework.[3]

Allowing non-regular R'. One direction for extending $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ is to allow *more expressive R' sets* for partial (co)directness. Let $\mathsf{PEP}^{\text{partial[DCFL]}}_{\text{codir}}$ and $\mathsf{PEP}^{\text{partial[Pres]}}_{\text{codir}}$ be like $\mathsf{PEP}^{\text{partial}}_{\text{codir}}$ except that R' can be any deterministic context-free $R'\in\mathsf{DCFL}(\Sigma)$ (respectively, any Presburger-definable $R'\in\mathsf{Pres}(\Sigma)$, i.e., a language whose Parikh image is a Presburger, or semilinear, subset of $\mathbb{N}^{|\Sigma|}$). Note that $R\in\mathsf{Reg}(\Sigma)$ is still required.

Theorem 6.1 (Undecidability). $\mathsf{PEP}^{\text{partial[DCFL]}}_{\text{codir}}$ *and* $\mathsf{PEP}^{\text{partial[Pres]}}_{\text{codir}}$ *are Σ^0_1-complete.*

Since both problems clearly are in Σ^0_1, one only has to prove hardness by reduction, e.g., from PCP, Post's Correspondence Problem. Let (Σ,Γ,u,v) be a PCP instance (where the question is whether there exists $x\in\Sigma^+$ such that $u(x)=v(x)$). Extend Σ and Γ with new symbols: $\Sigma'\stackrel{\text{def}}{=}\Sigma\cup\{1,2\}$ and $\Gamma'\stackrel{\text{def}}{=}\Gamma\cup\{\#\}$. Now define $u',v':\Sigma'^*\to\Gamma'^*$ by extending u,v on the new symbols with $u'(1)=v'(2)=\varepsilon$ and $u'(2)=v'(1)=\#$. Define now $R=12\Sigma^+$ and $R'=\{\tau 2\tau' \mid \tau,\tau'\in\Sigma^* \text{ and } |u(\tau\tau')|\neq|v(\tau\tau')|\}$. Note that R' is deterministic context-free and Presburger-definable.

Lemma 6.2. *The* PCP *instance (Σ,Γ,u,v) has a solution if and only if the* $\mathsf{PEP}^{\text{partial[Pres]}}_{\text{codir}}$ *and* $\mathsf{PEP}^{\text{partial[DCFL]}}_{\text{codir}}$ *instance $(\Sigma',\Gamma',u',v',R,R')$ has a solution.*

Combining directness and codirectness. Another direction is to allow *combining* directness and codirectness constraints. Formally, $\mathsf{PEP}_{\text{co\&dir}}$ is the problem of deciding, given Σ, Γ, u, v, and $R\in\mathsf{Reg}(\Sigma)$ as usual, whether there exists $\sigma\in R$ such that $u(\tau)\sqsubseteq v(\tau)$ and $u(\tau')\sqsubseteq v(\tau')$ for all decompositions $\sigma=\tau.\tau'$. In other words, σ is both a direct and a codirect solution.

[3] PEP is undecidable if we allow constraint sets R outside $\mathsf{Reg}(\Sigma)$ [2]. Other extensions, like $\exists x\in R_1:\forall y\in R_2 : u(xy)\sqsubseteq v(xy)$, for $R_1,R_2\in\mathsf{Reg}(\Sigma)$, have been shown undecidable [6].

Note that $\mathsf{PEP}_{\text{co\&dir}}$ has no R' parameter (or, equivalently, has $R' = \Sigma^*$) and requires directness and codirectness at all positions. However, this restricted combination is already undecidable:

Theorem 6.3 (Undecidability). $\mathsf{PEP}_{\text{co\&dir}}$ *is* Σ^0_1*-complete.*

Membership in Σ^0_1 is clear and we prove hardness by reducing from the Reachability Problem for length-preserving semi-Thue systems. The undecidability is linked to relying on *different* embeddings of $u(\sigma)$ in $v(\sigma)$ for the directness and codirectness. In contrast, for $\mathsf{PEP}^{\text{partial}}_{\text{dir}}$ we need to consider only the leftmost embedding of $u(\sigma)$ in $v(\sigma)$.

A semi-Thue system $S = (\Upsilon, \Delta)$ has a finite set $\Delta \subseteq \Upsilon^* \times \Upsilon^*$ of string rewrite rules over some alphabet Υ, written $\Delta = \{l_1 \to r_1, \ldots, l_k \to r_k\}$. The one-step rewrite relation $\to_\Delta \subseteq \Upsilon^* \times \Upsilon^*$ is defined as usual with $x \to_\Delta y \stackrel{\text{def}}{\Leftrightarrow} x = zlz'$ and $y = zrz'$ for some rule $l \to r$ in Δ and strings z, z' in Υ^*. We write $x \xrightarrow{m}_\Delta y$ and $x \xrightarrow{*}_\Delta y$ when x can be rewritten into y by a sequence of m (respectively, any number, possibly zero) rewrite steps.

The *Reachability Problem* for semi-Thue systems is "Given $S = (\Upsilon, \Delta)$ and two regular languages $P_1, P_2 \in \mathsf{Reg}(\Upsilon)$, is there $x \in P_1$ and $y \in P_2$ s.t. $x \xrightarrow{*}_\Delta y$?". It is well-known (or easy to see by encoding Turing machines in semi-Thue systems) that this problem is undecidable (in fact, Σ^0_1-complete) even when restricted to *length-preserving systems*, i.e., systems where $|l| = |r|$ for all rules $l \to r \in \Delta$.

We now construct a many-one reduction to $\mathsf{PEP}_{\text{co\&dir}}$. Let $S = (\Upsilon, \Delta)$, P_1, P_2 be a length-preserving instance of the Reachability Problem. W.l.o.g., we assume $\varepsilon \notin P_1$ and we restrict to reachability via an even and non-zero number of rewrite steps. With any such instance we associate a $\mathsf{PEP}_{\text{co\&dir}}$ instance $u, v : \Sigma^* \to \Gamma^*$ with $R \in \mathsf{Reg}(\Sigma)$ such that the following correctness property holds:

$$\begin{aligned} &\exists x \in P_1, \exists y \in P_2, \exists m \text{ s.t. } x \xrightarrow{m}_\Delta y \text{ (and } m > 0 \text{ is even)} \\ \text{iff } \ &\exists \sigma \in R \text{ s.t. } u(\tau) \sqsubseteq v(\tau) \text{ and } u(\tau') \sqsubseteq v(\tau') \text{ for all decompositions } \sigma = \tau\tau'. \end{aligned} \tag{CP}$$

The reduction uses letters like $\mathtt{a}$, $\mathtt{b}$ and $\mathtt{c}$ taken from Υ, and adds † as an extra letter. We use six copies of each such "plain" letter. These copies are obtained by priming and double-priming letters, and by overlining. Hence the six copies of $\mathtt{a}$ are $\mathtt{a}, \mathtt{a}', \mathtt{a}'', \overline{\mathtt{a}}, \overline{\mathtt{a}'}, \overline{\mathtt{a}''}$. As expected, for a "plain" word (or alphabet) x, we write x' and $\overline{x}$ to denote a version of x obtained by priming (respectively, overlining) all its letters. Formally, letting $\Upsilon_\dagger$ being short for $\Upsilon \cup \{\dagger\}$, one has $\Sigma = \Gamma \stackrel{\text{def}}{=} \Upsilon_\dagger \cup \Upsilon'_\dagger \cup \Upsilon''_\dagger \cup \overline{\Upsilon_\dagger} \cup \overline{\Upsilon'_\dagger} \cup \overline{\Upsilon''_\dagger}$.

We define and explain the reduction by running it on the following example:

$$\Upsilon = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\} \text{ and } \Delta = \{\mathtt{ab} \to \mathtt{bc},\ \mathtt{cc} \to \mathtt{aa}\}. \tag{S_{exmp}}$$

Assume that $\mathtt{abc} \in P_1$ and $\mathtt{baa} \in P_2$. Then $P_1 \xrightarrow{*}_\Delta P_2$ since $\mathtt{abc} \xrightarrow{*}_\Delta \mathtt{baa}$ as witnessed by the following (even-length) derivation π = "$\mathtt{abc} \to_\Delta \mathtt{bcc} \to_\Delta \mathtt{baa}$". In our reduction, a rewrite step like "$\mathtt{abc} \to_\Delta \mathtt{bcc}$" appears in the PEP solution σ as the letter-by-letter interleaving $\mathtt{a}\overline{\mathtt{b}}\mathtt{b}\overline{\mathtt{c}}\mathtt{c}\overline{\mathtt{c}}$, denoted $\mathtt{abc} \sqcup\!\sqcup \mathtt{bcc}$, of a plain string and an overlined copy of a same-length string.

Write $T_{\blacktriangleright}(\Delta)$, or just $T_{\blacktriangleright}$ for short, for the set of all $x \sqcup\!\sqcup y$ such that $x \to_\Delta y$. Obviously, and since we are dealing with length-preserving systems, $T_{\blacktriangleright}$ is a regular language, as seen by writing it as $T_{\blacktriangleright} = \left(\sum_{a\in\Upsilon} a\overline{a}\right)^*.\left\{l \sqcup\!\sqcup r \mid l \to r \in \Delta\right\}.\left(\sum_{a\in\Upsilon} a\overline{a}\right)^*$, where $\{l \sqcup\!\sqcup r \mid l \to r \in \Delta\}$ is a finite, hence regular, language.

$T_{\blacktriangleright}$ accounts for odd-numbered steps. For even-numbered steps like $\mathtt{bcc}\to_\Delta\mathtt{baa}$ in π above, we use the symmetric $\mathtt{b}\overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{a}}\mathtt{c}\overline{\mathtt{c}}$, i.e., $\mathtt{baa} \sqcup\!\sqcup \mathtt{bcc}$. Here too $T_{\blacktriangleleft} \stackrel{\text{def}}{=} \{y \sqcup\!\sqcup x \mid x \to_\Delta y\}$ is regular. Finally, a derivation π of the general form $x_0 \to_\Delta x_1 \to_\Delta x_2 \ldots \to_\Delta x_{2k}$, where $K \stackrel{\text{def}}{=} |x_0| = \ldots = |x_{2k}|$, is encoded as a solution σ_π of the form $\sigma_\pi = \rho_0\sigma_1\rho_1\sigma_2\ldots\rho_{2k-1}\sigma_{2k}\rho_{2k}$ that alternates between the encodings of steps (the σ_i's) in $T_{\blacktriangleright} \cup T_{\blacktriangleleft}$, and *fillers*, (the ρ_i's) defined as follows:

$$\sigma_i \stackrel{\text{def}}{=} \begin{cases} x_{i-1} \sqcup\!\sqcup x_i \text{ for odd } i\,, \\ x_i \sqcup\!\sqcup x_{i-1} \text{ for even } i\,, \end{cases} \quad \rho_0 \stackrel{\text{def}}{=} x_0'' \sqcup\!\sqcup \dagger''^K\,, \quad \rho_{2k} \stackrel{\text{def}}{=} x_{2k}'' \sqcup\!\sqcup \dagger''^K\,, \quad \rho_i \stackrel{\text{def}}{=} \begin{cases} \dagger'^K \sqcup\!\sqcup x_i' \text{ for odd } i\,, \\ x_i' \sqcup\!\sqcup \dagger'^K \text{ for even } i \neq 0, 2k\,. \end{cases}$$

Note that the extremal fillers ρ_0 and ρ_{2k} use double-primed letters, when the internal fillers use primed letters. Continuing our example, the σ_π associated with the derivation $\mathtt{abc}\to_\Delta\mathtt{bcc}\to_\Delta\mathtt{baa}$ is

$$\sigma_\pi = \underbrace{\mathtt{a}''\overline{\dagger''}\mathtt{b}''\overline{\dagger''}\mathtt{c}''\overline{\dagger''}}_{\mathtt{a}''\mathtt{b}''\mathtt{c}''\sqcup\!\sqcup\dagger''\dagger''\dagger''} \underbrace{\mathtt{a}\overline{\mathtt{b}}\mathtt{b}\overline{\mathtt{c}}\mathtt{c}\overline{\mathtt{c}}}_{\mathtt{abc}\sqcup\!\sqcup\mathtt{bcc}} \underbrace{\dagger'\overline{\mathtt{b}'}\dagger'\overline{\mathtt{c}'}\dagger'\overline{\mathtt{c}'}}_{\dagger'\dagger'\dagger'\sqcup\!\sqcup\mathtt{b}'\mathtt{c}'\mathtt{c}'} \underbrace{\mathtt{b}\overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{c}}\mathtt{a}\overline{\mathtt{c}}}_{\mathtt{baa}\sqcup\!\sqcup\mathtt{bcc}} \underbrace{\mathtt{b}''\overline{\dagger''}\mathtt{a}''\overline{\dagger''}\mathtt{a}''\overline{\dagger''}}_{\mathtt{b}''\mathtt{a}''\mathtt{a}''\sqcup\!\sqcup\dagger''\dagger''\dagger''}.$$

The point with primed and double-primed copies is that u and v associate them with different images. Precisely, we define

$$\begin{array}{lllll} u(a) = a, & u(a') = \dagger, & u(\dagger') = \dagger, & u(a'') = \varepsilon, & u(\dagger'') = \varepsilon, \\ v(a) = \dagger, & v(a') = a, & v(\dagger') = w_\Upsilon, & v(a'') = a, & v(\dagger'') = w_\Upsilon, \end{array}$$

where a is any letter in Υ, and where w_Υ is a word listing all letters in Υ. E.g., $w_{\{\mathtt{a},\mathtt{b},\mathtt{c}\}} = \mathtt{abc}$ in our running example. The extremal fillers use special double-primed letters because we want $u(\rho_0) = u(\rho_{2k}) = \varepsilon$ (while v behaves the same on primed and double-primed letters). Finally, overlining is preserved by u and v: $u(\overline{x}) \stackrel{\text{def}}{=} \overline{u(x)}$ and $v(\overline{x}) \stackrel{\text{def}}{=} \overline{v(x)}$.

This ensures that, for $i > 0$, $u(\sigma_i) \sqsubseteq v(\rho_{i-1})$ and $u(\rho_i) \sqsubseteq v(\sigma_i)$, so that a σ_π constructed as above is a direct solution. It also ensures $u(\sigma_i) \sqsubseteq v(\rho_i)$ and $u(\rho_{i-1}) \sqsubseteq v(\sigma_i)$ for all $i > 0$, so that σ_π is also a codirect solution. One can check it on our running example by writing $u(\sigma_\pi)$ and $v(\sigma_\pi)$ alongside:

$$\begin{array}{rl} \sigma_\pi = & \overbrace{\mathtt{a}''\overline{\dagger''}\mathtt{b}''\overline{\dagger''}\mathtt{c}''\overline{\dagger''}}^{\rho_0}\ \overbrace{\mathtt{a}\overline{\mathtt{b}}\mathtt{b}\overline{\mathtt{c}}\mathtt{c}\overline{\mathtt{c}}}^{\sigma_1}\ \overbrace{\dagger'\overline{\mathtt{b}'}\dagger'\overline{\mathtt{c}'}\dagger'\overline{\mathtt{c}'}}^{\rho_1}\ \overbrace{\mathtt{b}\overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{c}}\mathtt{a}\overline{\mathtt{c}}}^{\sigma_2}\ \overbrace{\mathtt{b}''\overline{\dagger''}\mathtt{a}''\overline{\dagger''}\mathtt{a}''\overline{\dagger''}}^{\rho_2} \\ \hline u(\sigma_\pi) = & \mathtt{a}\overline{\mathtt{b}}\mathtt{b}\overline{\mathtt{c}}\mathtt{c}\overline{\mathtt{c}}\ \ \dagger\overline{\dagger}\dagger\overline{\dagger}\dagger\overline{\dagger}\ \ \mathtt{b}\overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{c}}\mathtt{a}\overline{\mathtt{c}} \\ v(\sigma_\pi) = & \mathtt{a}\overline{\mathtt{abc}}\mathtt{b}\overline{\mathtt{abc}}\mathtt{c}\overline{\mathtt{abc}}\ \ \dagger\overline{\dagger}\dagger\overline{\dagger}\dagger\overline{\dagger}\ \ \mathtt{abc}\overline{\mathtt{b}}\mathtt{abc}\overline{\mathtt{c}}\mathtt{abc}\overline{\mathtt{c}}\ \ \dagger\overline{\dagger}\dagger\overline{\dagger}\dagger\overline{\dagger}\ \ \mathtt{b}\overline{\mathtt{abc}}\mathtt{a}\overline{\mathtt{abc}}\mathtt{a}\overline{\mathtt{abc}} \end{array}$$

There remains to define R. Since $\rho_0 \in \left(\Upsilon''\overline{\dagger''}\right)^+$, since $\sigma_i \in T_{\blacktriangleright}$ for odd i, etc., we let

$$R \stackrel{\text{def}}{=} \left(\Upsilon''\overline{\dagger''}\right)^+.T_{\blacktriangleright}^{\cap P_1}.\left(\dagger'\overline{\Upsilon'}\right)^+.\left(T_{\blacktriangleleft}.\left(\Upsilon'\overline{\dagger'}\right)^+.T_{\blacktriangleright}.\left(\dagger'\overline{\Upsilon'}\right)^+\right)^*.T_{\blacktriangleleft}^{\cap P_2}.\left(\Upsilon''\overline{\dagger''}\right)^+, \qquad (1)$$

where $T_{\blacktriangleright}^{\cap P_1} \stackrel{\text{def}}{=} \{x \sqcup y \mid x \rightarrow_\Delta y \wedge x \in P_1\} = T_{\blacktriangleright} \cap \{x \sqcup y \mid x \in P_1 \wedge |x| = |y|\}$ is clearly regular when P_1 is, and similarly for $T_{\blacktriangleleft}^{\cap P_2} \stackrel{\text{def}}{=} \{y \sqcup x \mid x \rightarrow_\Delta y \wedge y \in P_2\}$. Since $\sigma_\pi \in R$ when π is an even-length derivation from P_1 to P_2, we deduce that the left-to-right implication in (CP) holds.

We refer to the full version of this paper at arxiv.org/abs/1109.1691 for a proof that the right-to-left implication also holds, which concludes the proof of Theorem 6.3.

7 Concluding remarks

We introduced partial directness in Post Embedding Problems and proved the decidability of $\mathsf{PEP}_{\text{dir}}^{\text{partial}}$ by showing that an instance has a solution if, and only if, it has a solution of length bounded by a computable function of the input. This generalizes and simplifies earlier proofs for PEP and $\mathsf{PEP}_{\text{dir}}$. The added generality is non-trivial and leads to decidability for UCST, or UCS (that is, unidirectional channel systems) extended with tests [10]. The simplification lets us deal smoothly with counting or universal versions of the problem. Finally, we showed that *combining* directness and codirectness constraints leads to undecidability.

References

1. P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.
2. P. Chambart and Ph. Schnoebelen. Post Embedding Problem is not primitive recursive, with applications to channel systems. In *Proc. FST&TCS 2007*, vol. 4855 of *LNCS*, pages 265–276. Springer, 2007.
3. P. Chambart and Ph. Schnoebelen. Mixing lossy and perfect fifo channels. In *Proc. CONCUR 2008*, vol. 5201 of *LNCS*, pages 340–355. Springer, 2008.
4. P. Chambart and Ph. Schnoebelen. The ω-Regular Post Embedding Problem. In *Proc. FOSSACS 2008*, vol. 4962 of *LNCS*, pages 97–111. Springer, 2008.
5. P. Chambart and Ph. Schnoebelen. The ordinal recursive complexity of lossy channel systems. In *Proc. LICS 2008*, pp. 205–216. IEEE Comp. Soc. Press, 2008.
6. P. Chambart and Ph. Schnoebelen. Computing blocker sets for the Regular Post Embedding Problem. In *Proc. DLT 2010*, vol. 6224 of *LNCS*, pp. 136–147. Springer, 2010.
7. P. Chambart and Ph. Schnoebelen. Pumping and counting on the Regular Post Embedding Problem. In *Proc. ICALP 2010*, vol. 6199 of *LNCS*, pp. 64–75. Springer, 2010.
8. M. Fairtlough and S. S. Wainer. Hierarchies of provably recursive functions. In S. Buss, editor, *Handbook of Proof Theory*, chapter 3, pp. 149–207. Elsevier Science, 1998.
9. A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.
10. P. Jančar, P. Karandikar, and Ph. Schnoebelen. Unidirectional channel systems can be tested. In preparation, 2012.
11. A. Muscholl. Analysis of communicating automata. In *Proc. LATA 2010*, vol. 6031 of *LNCS*, pp. 50–57. Springer, 2010.
12. S. Schmitz and Ph. Schnoebelen. Multiply-recursive upper bounds with Higman's lemma. In *Proc. ICALP 2011*, vol. 6756 of *LNCS*, pages 441–452. Springer, 2011.